

CLAIMS

What is claimed is:

- 1 1. A method comprising:
 - 2 receiving a string of data symbols; and
 - 3 compressing the string of data into a fixed sized compressed data block
 - 4 having a plurality of compressed symbols and dictionary elements, the symbols
 - 5 and dictionary elements having a fixed length and a fixed offset.
- 1 2. The method of claim 1 wherein compressing the data comprises:
 - 2 dividing a first symbol into a first component and a second component;
 - 3 and
 - 4 comparing the first component with the dictionary elements.
- 1 3. The method of claim 2 further comprising compressing the first
 - 2 component to form a first tag if the first component matches a dictionary
 - 3 element.
- 1 4. The method of claim 3 wherein each symbol includes a tag to indicate a
 - 2 compression type.
- 1 5. The method of claim 3 further comprising storing the first component at a
 - 2 dictionary element if the first component does not match a dictionary element.
- 1 6. The method of claim 3 wherein compressing the data comprises:

2 dividing a second symbol into a second component and a second
3 component; and
4 comparing the second component with the dictionary elements.

1 7. A compression system:

2 a register to store a plurality of fixed length data symbols to be
3 compressed;
4 compression logic to compress each of the plurality of data symbols to
5 form a compressed symbol, the compressed symbols forming a compressed data
6 block having a fixed offset; and
7 a plurality of dictionary registers to store dictionary elements having a
8 fixed length.

1 8. The system of claim 7 wherein each symbol is divided into a first
2 component and a second component.

1 9. The method of claim 8 wherein the first and second components are
2 compressed into fixed length tags.

1 10. The method of claim 8 wherein the first and second components are
2 compressed into variable length tags.

1 11. The system of claim 8 wherein the first component is received at the

2 compression logic and encoded to form a tag.

1 12. The system of claim 11 further comprising a buffer to store the tag and
2 second component of each symbol as the compressed symbol.

1 13. The system of claim 8 wherein the compression logic comprises:
2 dictionary matching logic to determine if the first component matches a
3 dictionary element; and
4 constant match logic to determine if the second component has all ones or
5 all zeroes.

1 14. The system of claim 13 wherein the compression logic comprises an
2 encoder coupled to the match logic and the no match logic to encode the first
3 component to form a tag if the first component matches a dictionary element, has
4 all ones or zeroes.

1 15. A method comprising:
2 receiving a fixed offset compressed data block having a plurality of
3 dictionary elements and compressed symbols; and
4 decompressing each of the compressed symbols in parallel.

1 16. The method of claim 15 wherein each of the compressed symbols are
2 decompressed simultaneously.

1 17. The method of claim 15 wherein decompressing each of the compressed
2 symbols comprises:

3 analyzing a tag component within a compressed symbol; and

4 decompressing the compressed symbol to form a symbol based upon the
5 tag value.

1 18. The method of claim 17 wherein decompressing the compressed symbol
2 to form a symbol based upon the tag value comprises:

3 decoding the tag to form a matched component of the symbol; and

4 combining the matched component with an unmatched component within
5 the compressed symbol to form the symbol.

1 19. A decompression system comprising:

2 a plurality of decompression units to decompress a corresponding

3 compressed symbol within a compressed data block to generate an

4 uncompressed symbol, wherein the decompression units decompress the

5 compressed symbols in parallel.

1 20. The system of claim 19 wherein the compressed symbol comprises a tag
2 component and an unmatched symbol component.

1 21. The system of claim 20 wherein each decompression unit comprises logic

2 to decode the tag component of a compressed symbol to generate a matched
3 symbol component.

1 22. The system of claim 21 wherein each decompression unit combines a
2 matched symbol component with the unmatched symbol component to form an
3 uncompressed symbol.

1 23. A computer system comprising:
2 a central processing unit (CPU);
3 a cache memory coupled to the CPU having a plurality of compressible
4 cache lines to store additional data; and
5 a cache controller comprising compression logic to compress each of the
6 plurality of cache lines by compressing the data within a compressed cache line
7 into a fixed sized compressed data block having a plurality of offset compressed
8 symbols and dictionary elements, the symbols and dictionary elements having a
9 fixed length and fixed offset.

1 24. The computer system of claim 23 wherein the cache controller further
2 comprises decompression logic to decompress compressed symbols within a
3 compressed data block to generate uncompressed symbols.

1 25. The computer system of claim 24 wherein the decompression logic
2 decompresses the compressed symbols in parallel.

1 26. A computer system comprising:
2 a central processing unit (CPU);
3 a cache memory coupled to the CPU having a plurality of compressible
4 cache lines to store additional data;
5 a chipset, coupled to the CPU and the cache memory, including:
6 compression logic to compress each of the plurality of cache lines
7 by compressing the data within a compressed cache line into a fixed sized
8 compressed data block having a plurality of offset compressed symbols
9 and dictionary elements, the symbols and dictionary elements having a
10 fixed length and fixed offset; and
11 a main memory coupled to the chipset;

1 27. The computer system of claim 26 wherein the chipset further comprises
2 decompression logic to decompress compressed symbols within a compressed
3 data block to generate uncompressed symbols.

1 28. A method comprising:
2 receiving a fixed offset compressed data block having a plurality of
3 dictionary elements and compressed symbols; and
4 decompressing a randomly accessed and a first compressed symbol
5 within the compressed data block.

1 29. The method of claim 28 wherein decompressing the first compressed
2 symbol comprises:
3 analyzing a tag component within a compressed symbol; and
4 decompressing the compressed symbol to form a symbol based upon the
5 tag value.